REMARKS

Applicant respectfully requests reconsideration of the present application in view of the amendments set forth above and the remarks below.

Claims 13 -22 are pending in the application. Claims 13-22 are rejected. Claim 13 is amended herein.

Claim 23 is new, support for which can be found throughout the application as-filed including, for example, at FIG. 2B and page 6, lines 20 - 33 of the specification.

The Prior Art Rejections under 35 U.S.C. § 101:

On page 2 of the Office Action, the Examiner rejects Claim 1 under 35 U.S.C. § 101 because the claimed invention is "directed to non-statutory subject matter." Claim 1 was previously withdrawn and Applicants therefore assume that the Examiner refers to Claim 13.

Claim 13 has been amended to recite:

> A computer-implemented method of scheduling processing in a hardware threaded circuit, comprising:
>
> in a processor, receiving inputs corresponding to unthreaded processing of an application;
>
> receiving and storing in a memory information including processing element resources, a number of processing elements, and a window size corresponding to a number of downstream processing states to be examined; and
>
> generating a hardware threaded schedule for processing the application with at least first and second one of the processing elements being interconnected to enable dynamic resource sharing.

Accordingly, Applicants respectfully request removal of the rejection under 35 U.S.C. § 101.

The Prior Art Rejections under 35 U.S.C. § 102(b):

Claims 13-22 are rejected under 35 U.S.C. § 102(b) as being anticipated by US Patent No. 5,519,867 to Moeller et al. (hereinafter "Moeller").

Applicants respectfully point out that the Examiner's reliance on Moeller in the present Office Action is substantially the same as that of US Patent No. 6,684,261 to Orton et al. (hereinafter "Orton"), which the Examiner relied on to reject the claims in a prior office action dated December 26, 2008 (the "Prior Office Action"). In particular, Applicants point out that Moeller and Orton are related to US Patent 5,379,432 to Orton et al. (the '432 patent) in that (1) Moeller is subject to a terminal disclaimer such that the term of Moeller shall not extend beyond the expiration date of the '432 patent, and; (2) Orton is a continuation of the '432 patent. Moeller is also a co-inventor on Orton and the '432 patent.

Moreover, the passages that the Examiner relies upon in Moeller to reject the claims in the present Office Action are virtually identical (except for some minor typographical and formatting differences) to passages in Orton and the Examiner has used the corresponding passages in Orton to similarly reject the claims in the Prior Office Action. For example, the Examiner relies on FIG. 4 and col. 6 line 41 to col. 7, line 59 of Moeller to teach "receiving inputs corresponding to unthreaded processing of an application" in claim13 (see page 3 of the Office Action) and in the Prior Office Action relied on Orton at FIG. 4 and col. 7, line 50 to col. 8, line 31 to teach these features of claim13. **FIG. 4 in Moeller and FIG. 4 in Orton are the same and so are the respective cited texts** (as are FIG. 11 and the remaining figures).

Applicants therefore submit that the Applicants' remarks that obviate the Examiner's claim rejections in the Prior Office Action (which have been reiterated herein) similarly obviate the Examiner's claim rejections in the present Office Action. Applicants also assert that citing the corresponding passages in this way in two consecutive office actions has inflicted an unnecessary burden both in time and cost on the Applicants.

Applicants traverse the Examiner's rejection of Claims 13-22 under 35 U.S.C. § 102(b) at least because Moeller does not disclose "a computer-implemented method of scheduling processing in a hardware threaded circuit, comprising . . . in a processor, receiving inputs corresponding to unthreaded processing of an application; receiving and storing in a memory information including processing element resources, a number of processing elements, and a

window size corresponding to a number of downstream processing states to be examined; and generating a hardware threaded schedule for processing the application with at least first and second one of the processing elements being interconnected to enable dynamic resource sharing," as set forth in Claim 13.

The present invention is directed to a hardware threading mechanism providing temporary borrowing of unutilized pipeline stages from processing elements to boost throughput performance and/or to reduce the power consumption of tasks running on active processing elements (see application as-filed, specification at page 5, line 31 - page 6, line 3). For example, the hardware threading may include dynamic borrowing of available processing resources between interconnected processing elements (see application as-filed, specification at page 2, lines 5 - 13, and claim 13).

The Examiner relies on Moeller at FIG. 4 and col. 6, line 41 through col. 7, line 59 to teach "receiving inputs corresponding to unthreaded processing of an application." While these passages may describe "thread classes 404 for enabling an application to access in an object-oriented manner operating system services to spawn, control, and obtain information relating to threads," such a generic description of thread classes for object-oriented thread control in a native procedural operating system environment in no way discloses the claimed "receiving inputs corresponding to unthreaded processing of an application."

The Examiner relies on Moeller at FIGS. 4 and 11, and col. 32, line 6 to col. 34, line 24 and background to teach "receiving information including processing element resources, a number of processing elements, and a window size corresponding to a number of downstream processing states to be examined," as set forth in claim 13. While these passages may describe a various scheduling and faults classes, nowhere is there mention of "receiving information including processing element resources, a number of processing elements, and a window size corresponding to a number of downstream processing states to be examined," as in claim 13.

In particular, Moeller discloses generic scheduling classes such as a TThreadSchedule concrete base class that embodies a scheduling behavior of a class, a TIdleThreadSchedule concrete subclass for threads running during system idle, a TServerSchedule concrete subclass for server threads, a TUserInterfaceSchedule concrete subclass for user interface handling, a TApplicationSchedule class to support long-running threads, and a TPseduoRealTimeThreadSchedule to indicate thread urgency. See Moeller at col. 32, lines 6 - 55. Moeller, therefore, describes various scheduling of base classes and subclasses that may be of use in an object-oriented system wrapper for a procedural operating system.

More to this point, it is clear than none of the above-mentioned classes contemplate "receiving information including processing element resources, a number of processing elements, and a window size corresponding to a number of downstream processing states to be examined," as claimed. Advantageously, the claimed invention may traverse downstream states in a schedule to look ahead by one or more states based on the window size (see application as-filed, e.g. the specification at page 14, line 31 through page 15, line 3). Future operations can be rescheduled earlier (such as within a current state) by maximizing the usage of underutilized resources.

Moreover, Moeller does not describe "generating a hardware threaded schedule for processing the application with at least first and second one of the processing elements being interconnected to enable dynamic resource sharing", as set forth in claim 13. The claimed invention may use such information to schedule more operations within a current state of an unthreaded schedule (see application as-filed, specification at page 15, lines 5 - 7). For example, if a next state can be scheduled within a current state, such as when next state can be scheduled by borrowing an unused resource, the two states are scheduled together and the states are said to be threaded together (see application as-filed, specification at page 15, lines 8 - 10).

The Examiner relies on Moeller at FIG. 4 and the text at col. 10, line 60 to col. 12, line 67, and col. 34, line 52 to col. 37, line 11 to teach "generating a hardware threaded schedule for processing the application with at least first and second one of the processing elements being

interconnected to enable dynamic resource sharing," as set forth in claim 13. While these passages may describe scheduling classes and thread classes, nowhere is there mention of the claimed features in these passages or in any other portion of Moeller. For example, Moeller at col. 20, line 65 - col. 21, line 3 states:

> "TThreadHandle is a concrete class that represents a thread entity in the system. It provides the methods for controlling and determination about the thread. It also provides the mechanism for spawning new threads in the system. Control operations include killing, suspending/resuming, and doing a death watch on it."

Therefore, while Moeller may describe basic thread functionality, such as spawning, killing, and suspending/resuming threads, it does not describe or contemplate the claimed "generating a hardware threaded schedule for processing the application with at least first and second one of the processing elements being interconnected to enable dynamic resource sharing." For example, nowhere does Moeller describe the claimed "at least first and second one of the processing elements" or "being interconnected to enable dynamic resource sharing."

Furthermore, Applicants submit that Moeller is concerned with *distinctly different* problems than with the problems addressed by the claimed invention. In particular, Moeller is concerned with an object-oriented system wrapper for allowing object-oriented calls within a native procedural operating system environment (see Detailed Description of Moeller at col. 4, line 65 - col. 5, line 8). Clearly, these problems are different than the problems associated with scheduling processing in a hardware threaded circuit. Therefore, Applicants submit that one of ordinary skill in the art faced with the problems addressed by the claimed invention would not look toward Moeller for a solution.

Accordingly, Applicants submit that Moeller does not describe "a computer-implemented method of scheduling processing in a hardware threaded circuit, comprising . . . in a processor, receiving inputs corresponding to unthreaded processing of an application; receiving and storing in a memory information including processing element resources, a number of processing elements, and a window size corresponding to a number of downstream processing states to be

examined; and generating a hardware threaded schedule for processing the application with at least first and second one of the processing elements being interconnected to enable dynamic resource sharing," as set forth in claim 13. Therefore, Applicants submit that claim 13 is patentable over Moeller and request withdrawal of the art rejections. For at least the same reasons, Applicants submit that claims 14 – 17 are also patentable over Moeller and request withdrawal of the art rejections.

Applicant submits that dependent claims contain additional subject matter that further distinguishes over the art of reference.

With regard to claim 14, Moeller fails to anticipate "synthesizing the hardware threaded schedule to an Application Specific Circuit (ASC)."

With regard to claim 15, Moeller fails to anticipate "synthesizing the hardware schedule to maximize throughput."

With regard to claim 16, Moeller fails to anticipate "synthesizing the hardware threaded schedule to reduce power consumption."

With regard to claim 17, Moeller fails to anticipate "receiving resource constraint information for the processing elements."

With regard to independent claim 18, Applicants submit that Moeller does not describe or contemplate "a hardware threaded circuit system, comprising: . . . a plurality of processing elements coupled to the task manager, wherein first and second ones of the plurality of processing elements are interconnected for hardware threaded processing to enable dynamic borrowing of processing resources associated with the second one of the plurality of processing elements by the first one of the plurality of processing elements," as recited in claim 18. The Examiner alleges that Moeller teaches these features at FIG. 4, col. 10, line 60 to col. 12, line 67, and col. 34, line 52, to col. 37, line 11. As described above, these passages describe threads and

thread classes for providing an object-oriented interface to a procedural operating system, such as the Mach micro-kernel developed by Carnegie Melon University (see Moeller at col. 5, line 55).

Accordingly, Applicants submit that Moeller does not describe "a hardware threaded circuit system, comprising: . . . a plurality of processing elements coupled to the task manager, wherein first and second ones of the plurality of processing elements are interconnected for hardware threaded processing to enable dynamic borrowing of processing resources associated with the second one of the plurality of processing elements by the first one of the plurality of processing elements," as recited in claim 18. Therefore, Applicants submit that claim 18 is patentable over Moeller and request withdrawal of the art rejections. For at least the same reasons, Applicants submit that claims 19 – 22 are also patentable over Moeller and request withdrawal of the art rejections.

Applicant submits that dependent claims 19 - 22 contain additional subject matter that further distinguishes over the art of reference.

With regard to claim 19, Moeller fails to anticipate "the circuit maximizes throughput."

With regard to claim 20, Moeller fails to anticipate "the circuit reduces power consumption compared to a non-threaded processing for substantially similar system wait times."

With regard to claim 21, Moeller fails to anticipate "the first and second processing elements each include a first type of resource and a second type of resource and a multiplexer such that the interconnection includes at least one input signal being provided to the first type of resource in the first and second processing elements." In fact, Moeller fails to even mention a multiplexor and Applicants submit that it is hard to imagine how Moeller could teach such a claimed feature to one of ordinary skill in the art.

With regard to claim 22, Moeller fails to anticipate "the interconnection includes a connection from an output of the second processing element first type of resource to the first processing element."

New claim 23 recites "the method of claim 13, wherein the at least first and second one of the processing elements are multiplexed." Applicants submit that Moeller fails to teach "at least first and second one of the processing elements are multiplexed," at least because Moeller describes an object-oriented wrapper for a procedural operating system and fails to even mention multiplexed.

It is believed that all of the pending claims have been addressed. However, the absence of a reply to a specific rejection, issue, or comment does not signify agreement with or concession of that rejection, issue, or comment. In addition, because the arguments made above may not be exhaustive, there may be reasons for withdrawing the prior art cited with regards to any or all pending claims (or other claims) that have not been expressed. Finally, nothing in this paper should be construed as intent to concede any issue with regard to any claim, except as specifically stated in this paper, and the amendment of any claim does not necessarily signify concession of unpatentability of the claim prior to its amendment.

Applicants submit that the entire application is now in condition for allowance. Such action is respectfully requested at the Examiner's earliest convenience.

The Examiner is respectfully invited to telephone the undersigning attorney if there are any questions regarding this Amendment or this application.

The Assistant Commissioner is hereby authorized to charge payment of any additional fees associated with this communication or credit any overpayment to Deposit Account No. 500845, including but not limited to, any charges for extensions of time under 37 C.F.R. §1.136.

Dated: September 11, 2009

Respectfully submitted,

DALY, CROWLEY, MOFFORD & DURKEE, LLP

By: _____

Steven M. Cohen
Reg. No. 59,503
Attorney for Applicant(s)
354A Turnpike Street - Suite 301A
Canton, MA 02021-2714
Tel.: (781) 401-9988, Ext. 126
Fax: (781) 401-9966
smc@dc-m.com

128111